Virginia Tech ◆ ECE/CS 5565, Network Architecture and Protocols I ◆ Fall 2024

Project 4: Exploring TCP with Wireshark[1]

Introduction

In this project, you will examine the TCP transport protocol using a packet trace and Wireshark. Your work on this project consists of three parts.

- Part 1: Explore the basics of TCP
- Part 2: Explore the operation of TCP
- Part 3: Submit a project report covering your work

Before starting on the project, read through this assignment to know how work in different parts fits together. Also, review the class syllabus, especially sections on "Assessments," "Grading," "Late and Missed Work," "Grading Questions," and "Graduate Academic Integrity."

Capturing the Trace File

This project will use a packet trace of a file transfer from a local client to a remote server using HTTP and TCP. The remote server provides a web page that prompts for the name of a file stored on the local client. The web page then causes the transfer of the specified file from the client to the remote HTTP server using the HTTP POST[2] method. Wireshark is run during this transaction to obtain a trace of TCP traffic to and from the local client computer.

You may use the provided packet capture file (**project4_tcp.pcapng**) or you can capture your own trace file using the procedure in Appendix A of this project report. You are encouraged to capture your own trace file, but you may use the provided trace file if this is not feasible. Even if you use the provided trace file, review the steps in Appendix A to understand the conditions under which the trace file was obtained.

Overview of the Trace File

This section briefly describes the trace file. Note that some specifics may vary if you captured your own trace file, but the general structure of the trace should be the same. Observe the following aspects of the trace file.

- The trace begins with the TCP three-way handshake to establish the connection. The client sends the first SYN segment to the server at IP address 128.119.245.12.
- The HTTP exchange begins after the three-way handshake. The client sends the HTTP POST method to the server. The HTTP POST method will include information about the browser and will transfer the full 150-KB file to the server. However, the file is too large to fit in one segment, so multiple TCP segments are used to carry the HTTP POST method including the file to the server.

---

[1] This project is based in part on "Wireshark Lab: TCP v8.1," a supplement to *Computer Networking: A Top-Down Approach*, 8th edition, by J. F. Kurose and K. W. Ross, 2022 (https://gaia.cs.umass.edu/kurose_ross/index.php). The project uses resources generously provided by the Manning College of Information & Computer Science at the University of Massachusetts Amherst (https://www.cics.umass.edu/).

[2] See https://en.wikipedia.org/wiki/POST_(HTTP) on Wikipedia for more information on the HTTP POST method.

- Observe that the segments carrying data to the server include the TCP PSH option to tell TCP on the server to deliver this information immediately to the HTTP server application.
- The data transfer for the POST method is completed in a later segment. In Wireshark the intermediate segments used to carry POST information indicate "Reassembled PDU in frame: *N*" where frame *N* is the final frame carrying POST information. Wireshark marks this final frame as an HTTP frame instead of as a TCP frame.
- The HTTP POST transaction ends with the "HTTP/1.1 200 OK" message sent from the server to the client to acknowledge that it has successfully received and processed the POST method.
- There will be numerous TCP ACK segments in the trace. These are sent in both directions, but are mostly from the server to the client since TCP sends more data from the client to the server than in the reverse direction.
- You may see the client sending a "GET /favico.ico HTTP/1.1" request to the server and the server responding with an "HTTP/1.1 404 Not Found" message. Some browsers ask the server for an icon for the page, but this server does not have one.

Part 1: Explore the Basics of TCP

Answer the following questions based on either the provided trace file or the trace file that you captured.

1.1. Did you use the provided trace file or your own capture for the analysis?

1.2. What is the IP address and TCP port number used by the client computer?

1.3. What is the IP address and TCP port number used by the server computer?

1.4. What is the actual TCP sequence number (the raw value in Wireshark) specified by the client in its SYN message? Be sure to specify the raw sequence number and not the relative sequence number.

1.5. What is the actual TCP acknowledgement number (the raw value in Wireshark) specified by the server in the SYN, ACK message sent to the client? How is this value determined by the server?

1.6. What is the TCP sequence number (raw value) specified by the client in the TCP segment carrying the header of the HTTP POST method? Note that this is the segment containing the text "POST /wireshark-labs/…" and not the final segment carrying HTTP POST data. How many bytes are contained in the TCP payload for this TCP segment?

1.7. The segment discussed in the previous question is the first segment in the data transfer from the client to the server of the TCP connection. Considering this, answer these questions. For time, use the relative time values in Wireshark.

1.7.a. At what time was the first segment (the one with the "POST /wireshark-labs/…" text) sent?

1.7.b. At what time was the ACK from the server for this first data segment received by the client?

1.7.c. What is the RTT based on the first data segment sent and acknowledged?

1.7.d. Consider the second data segment sent from client to server. What is the RTT considering this data segment and the ACK from the server?

1.8. What is the minimum TCP window size advertised by the server to the client?

1.9. Are there any retransmitted segments in the packet trace? How did you determine this?

1.10. What is the data throughput for the transfer of the full POST method? Give your result in both bytes per second and bits per second. Show how you calculated the throughput values.

1.11. Briefly explain how the TCP connection is closed. Refer to specific TCP segments.

Part 2: Exploring the Operation of TCP

This part of the project explores the operation of TCP using the Sequence Numbers (Stevens) graphing tool in Wireshark. Perform the following steps.

1) In the Wireshark frame display window, select the SYN segment sent by the client to the server that begins to establish the TCP connection.

2) Select the Statistics > TCP Stream Graphs > Time Sequence (Stevens) menu. A graph is displayed. Be sure that the text at the top of the graph show that this is for client IP address ➔ server IP address (128.119.245.12:80). If the direction is server IP address ➔ client IP address, click the "Switch Direction" button. The graph should look something like what is shown in Figure 1. The dots in the graph represent different segments. You can click on a dot to get more information.

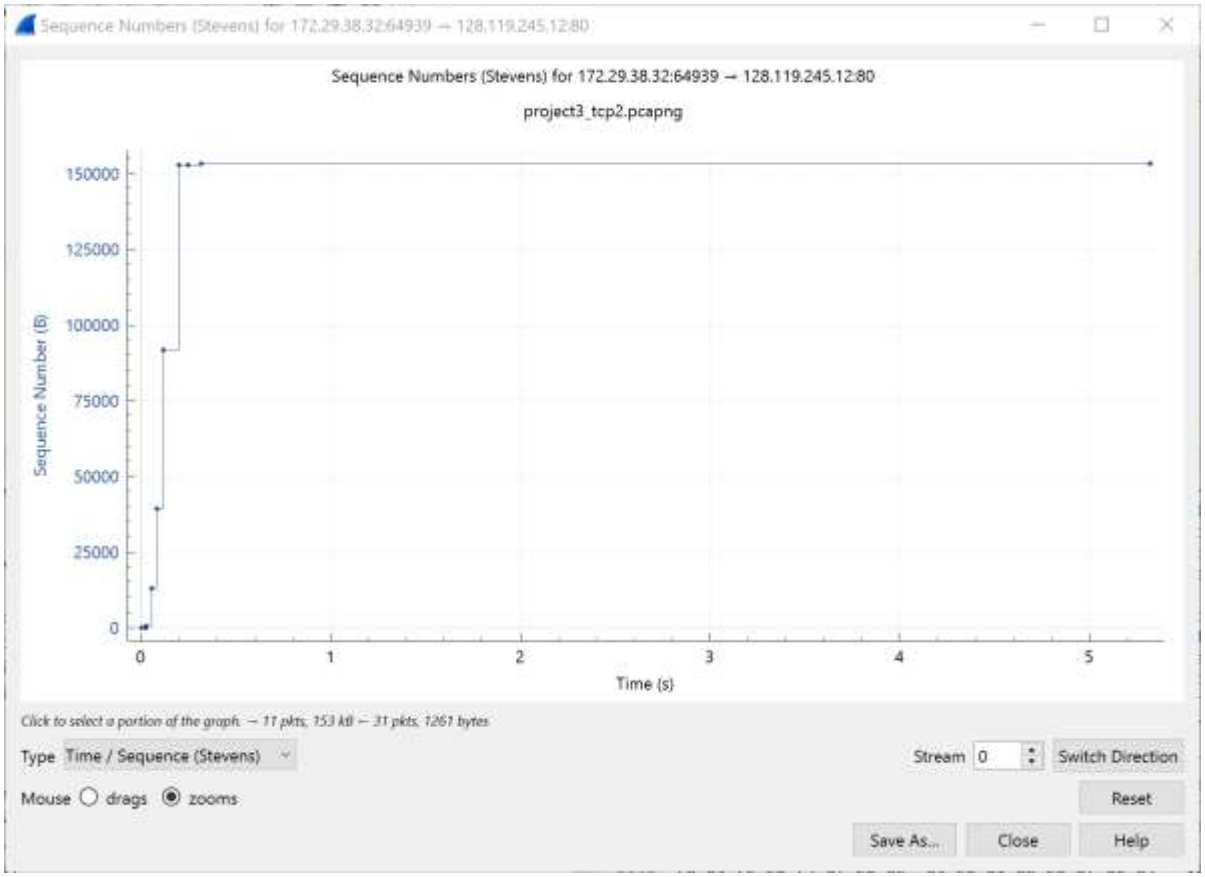

Figure 1. Wireshark graph of sequence numbers for client to server without zooming.

3) Select the Mouse Zooms option in the graph and zoom in to expand the part of the graph that is interesting in that the sequence number is changing. Zoom by dragging over to highlight the portion of the graph of interest. Click the "Reset" button to start over. The graph should look something like what is shown in Figure 2.

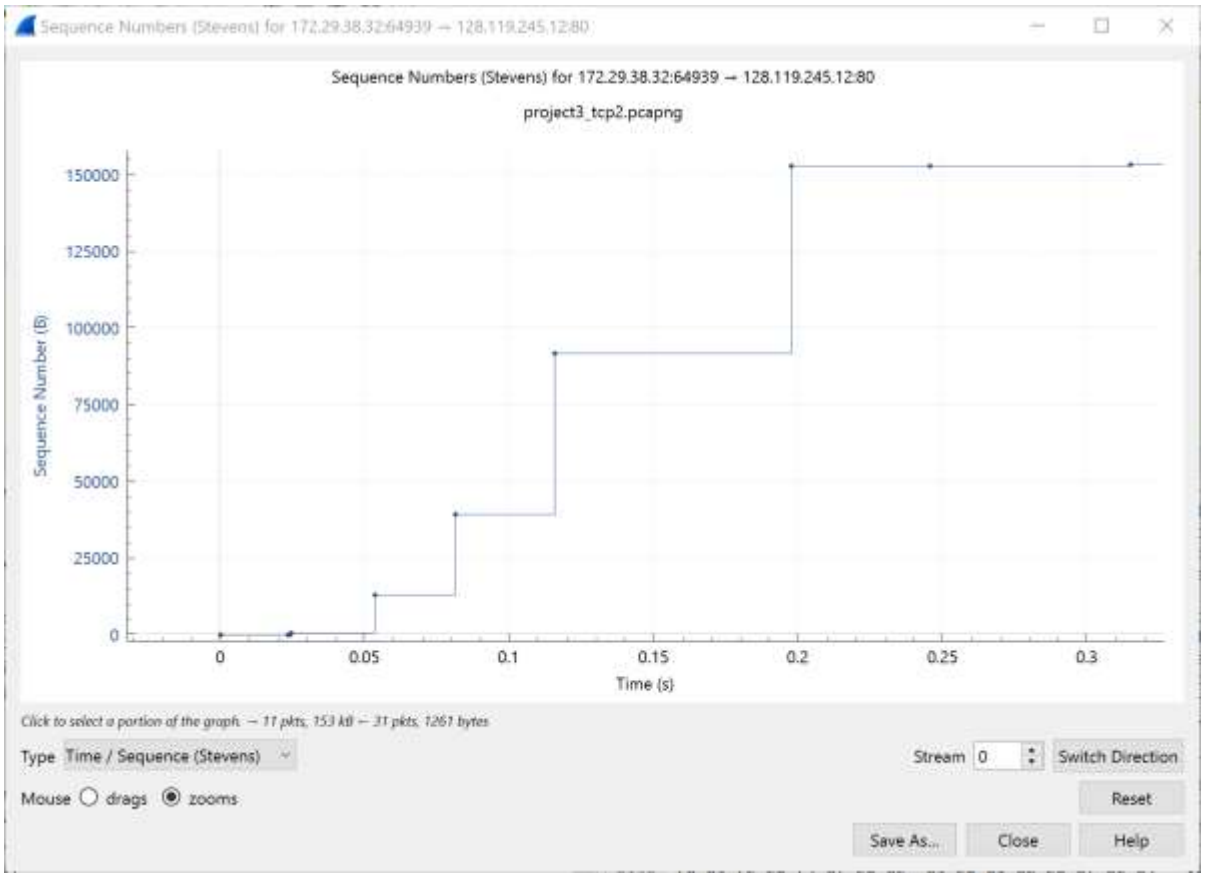

Figure 2. Wireshark graph of sequence numbers for client to server with zooming.

Answer the following questions.

2.1. Zoom into the Sequence Numbers (Stevens) window as described above. Make a screen capture of the graph for inclusion in the report.

2.2. What is the change in sequence numbers for each of the first four "steps" in the graph?

2.3. During this time, is TCP in the slow-start phase, congestion avoidance phase, or some other phase? What is the basis for your answer?

Part 3: Submission of the Report

Create and submit a written report with the content described below. Your full name, Virginia Tech PID, Virginia Tech email address, and assignment name ("ECE/CS 5565 Project 4") should appear at the top of the first page. Do *not* include your Student ID number.

Provide answers to all questions above from Part 1 (1.1-1.11) and Part 2 (2.1-2.3). In your report, provide three headings. The first two headings for Sections 1 and 2 should correspond to the headings for Parts 1 and 2 in the assignment. Under each section heading, provide answers for the questions from

each part of the assignment. Include the question numbers and answer all questions in order. All answers should be concise and clear.

Add a third section, "Section 3. Summary." Briefly indicate problems, if any, with the assignment. You must give some information, even if to say you encountered no problems.

Your report should be submitted as a single PDF file with the following file name:

*YourLastName_YourFirstName*_P4.pdf

Note that "YourLastName" is your last or family name as used by Virginia Tech and that "YourFirstName" is your first or given name as used by Virginia Tech. Submit the report in the Assignments section of the class Canvas site by the due date.

Honor System Expectations

Your work on this project and your submission should be your own. You may consult with others about how to use Wireshark. You are encouraged to ask such questions and provide responses to such questions using the "Project 4" topic in the Discussions section of the class Canvas site. **You are not to collaborate with others on the actual analysis of the trace file, providing the information for the report, or on writing the report. Such collaboration will be considered a violation of Virginia Tech's Graduate Honor Code.** Please review the section on "Graduate Academic Integrity" in the course syllabus available on the class Canvas site before you begin work on this project.

Grading Rubric

Your project will be graded using the following rubric as a guide. The maximum score is 100 points.

Project Criterion	Attributes of Strong Work	Attributes of Medium Work	Attributes of Weak Work	Maximum Points
Part 1 – Exploring the basics of TCP	All questions are answered correctly and are complete, specific, clear, and concise. (55-60 points)	There are some minor errors or omissions. Explanations are correct but incomplete or overly verbose. (40-54 points)	There are significant errors or omissions. Multiple explanations are incorrect or vague. (0-39 points)	60
Part 2 – Exploring the operation of TCP	All questions are answered correctly and are complete, specific, clear, and concise. The image of the graph from Wireshark is correct and clear. (25-30 points)	There are some minor errors or omissions. Explanations are correct but incomplete or overly verbose. The graph is included but is not clear. (15-24 points)	There are multiple omissions or other errors. Many of the explanations are incorrect or unclear. (0-14 points)	30
Overall presentation and submission	Work is clearly presented and organized with headings in the correct order. Text and the image are clear and clearly labeled. The brief summary section is provided. Submission instructions were followed. (9-10 points)	Work is mostly clear but with some lack of clarity in writing and/or the image. Organization is adequate, but not completely aligned with directions given. The summary section is missing. (6-8 points)	Work is hard to follow. Did not follow instructions for organizing the report. Did not fully follow submission instructions. The summary section is missing. (0-5 points)	10
TOTAL POINTS				100

Appendix A: Capturing a TCP Packet Trace

A text file, **alice.txt**, is provided for the file transfer. This is a 150-KB file that contains the full text of *Alice in Wonderland* by Lewis Carroll. Follow these steps to capture the packet trace for this project.

1) From a web browser, go to: https://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html. You should see something very similar to the web page in Figure A-1.

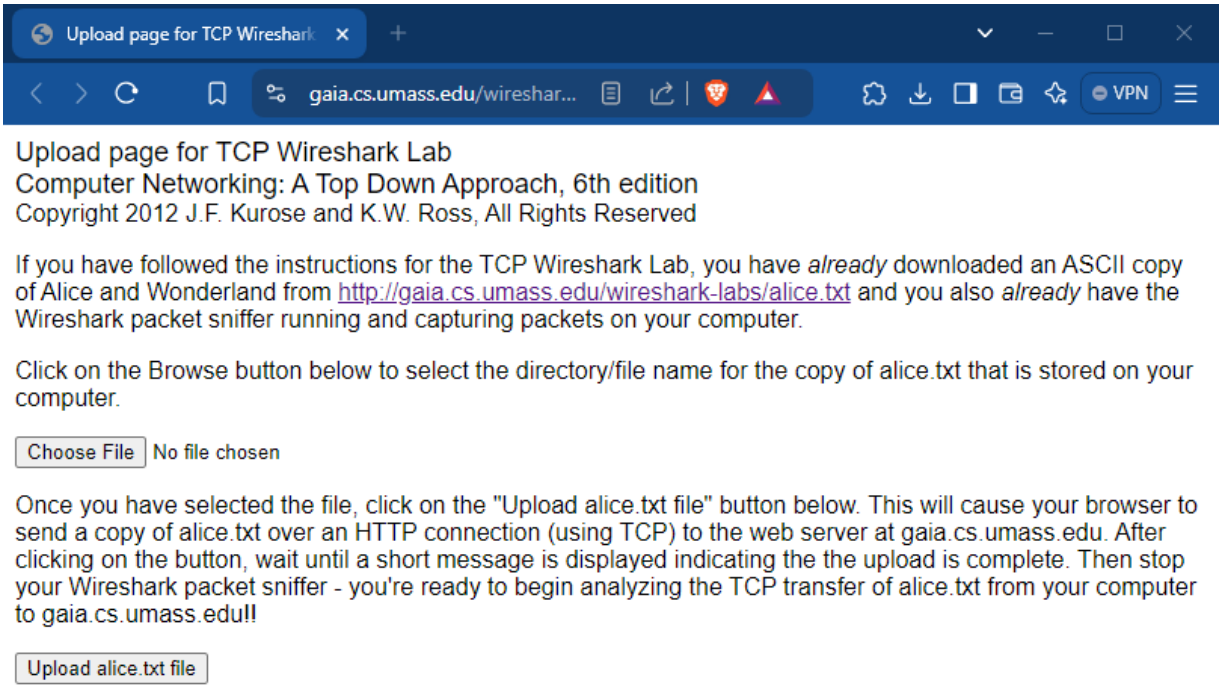

Figure A-1. Web page for file upload.

2) Use the "Choose File" button on the web page to select the **alice.txt** file on your computer.

3) Start the Wireshark packet capture after clicking the "Choose File" button using these steps.
   a) Open the Wireshark application.
   b) Select the Capture > Options… menu.
   c) In the "…using this filter" entry box enter: **tcp port 80**
   d) From the bottom of the Capture Options screen in Wireshark, confirm the interface that is active. It is likely something like "Wi-Fi." Do not consider "Adapter for loopback traffic capture." You should see something like the image shown in Figure A-2.
   e) Right click on the active interface and click "Start Capture." Wireshark will now show packets being captured. It is likely that no packets will be captured until the next step.

Figure A-2. Image from Wireshark Capture Options screen.

4) In your web browser, click the “Upload alice.txt file” button. Note that this button must be clicked after the Wireshark capture has been stared in the previous step. Depending on your browser, you may get a warning about the information not being secure. This is because the transfer is done with HTTP and not HTTPS. We want HTTP so that the contents of the transfer in the packet trace are not encrypted. A confirmation message will appear on the web page when the transfer is complete.

5) Wireshark should now show captured packets. Wait about 15 seconds after the confirmation message in the web browser to be sure that the packet capture has ended. You are likely to see a TCP [ACK] segment, a TCP [FIN, ACK] segment, and a TCP [ACK] segment as the last three captured packets in the Wireshark window as shown in Figure A-3. After waiting to be sure that the packet capture has ended, stop the packet capture. This can be done by pressing the red square in the tool bar or selecting the Capture > Stop menu.

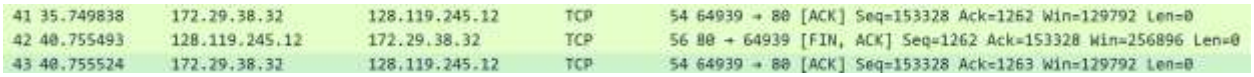

Figure A-3. Image from the Wireshark packet capture showing the last three captured frames.

6) Save the packet trace using the File > Save menu in Wireshark.

This completes the packet trace capture. You can close your browser. You can keep Wireshark open to explore the packet trace or come back to it later by opening the saved file.